# ORA SOFTWARE - USER MANUAL

**Disclaimer:** This software is provided "AS IS" without warranty of any kind. This is developmental code, and we make no pretension as to it being bug-free and totally reliable. Use at your own risk. We will accept no liability for any damages incurred through the use of this software.
Use of ORA is free, and the program is open source.

**How to cite:** If you publish results obtained in part by using ORA, then we require that you acknowledge this by citing the program as follows:
**Sardu A [1], Treu L [2], Campanaro S [3]** (2014) Transcriptome structure variability in Saccharomyces cerevisiae strains determined with a newly developed assembly software. BMC Genomics. 15:1045.
doi: 10.1186/1471-2164-15-1045.
1 - Department of Biology, University of Fribourg, Chemin du Musée 10, CH-1700 Fribourg FR, Switzerland
2 - Department of Environmental Engineering, Technical University of Denmark, Anker Engelunds Vej 1 Building 101A 2800 Kgs. Lyngby, Denmark
3 - Department of Biology, University of Padua, Via Ugo Bassi 58/b, Padova 35131, Italy

**Contact:** for information please contact Alessandro Sardu (Alessandro.sardu@unifr.ch), Stefano Campanaro (stefano.campanaro@gmail.com)

## FOR IMPATIENT PEOPLE

Download the software from "https://sourceforge.net/projects/transcriptomeassemblyora/files/" saving all the files ("launcher.pl" "overlapping_reads_assembler_v2.pl" "seconda_parte.pl" "terza_parte.pl" "quarta_parte.pl" "estrapola_chr.pl") into the same folder.
ORA requires (1) an ORDERED sam file obtained aligning reads on the reference genome, (2) a file with the name and the size of the chromosomes and (optional but strongly recommended) (3) a "gff" file with the annotation of the reference genome having a format compatible of that reported in the SGD.
Decompress the sam file using gunzip command:

```
user@machine:~/folder$ gunzip name_of_the_sam_file.gz
```

Run the software using

```
user@machine:~/folder$ ./launcher.pl file_sam_odered.sam chroms_name_and_size.txt annotation_file.gff
```

… and follow the instruction provided by the software step by step.

## RELEVANT NOTES

1-ORA software function on any standard Linux-based environment, it requires perl and a relevant amount of physical memory; it requires approximately 5-10 Gb of RAM memory for the assembly of a "typical" transcriptome like that of *Saccharomyces cerevisiae* sequenced with 20-30 million of reads aligned on the reference genome.

2-ORA is a software for reference-based transcriptome reconstruction and for this reason it requires a reference genome where the reads obtained from the RNA-seq process were aligned before the transcriptome reconstruction process.

3-To align the reads on the reference genome you can use a software like PASS (Campagna et al., 2009) (http://pass.cribi.unipd.it/cgi-bin/pass.pl) or Bowtie (Langmead et al., 2009) (http://bowtie-bio.sourceforge.net/index.shtml) or any other software generating an output in sam format (a test file is available "saccharomyces_cerevisiae_3Mreads_ordered.sam.gz"). If you are analyzing lower eukaryotes we suggest to use a software that can manage spliced reads, this allows introns identification.
The SAM output obtained from reads alignment has to be ordered using for example the samtools (Li et al., 2009) (http://samtools.sourceforge.net/).

4-Introns identification requires that spliced reads were identified by the software used for short-reads alignment, if this software cannot find and align spliced reads, introns cannot be identified and are not reported by ORA in the final output.

5-ORA requires a file with the name and the size of the chromosomes of the reference genome (a test file is available "saccharomyces_cerevisiae_S288c_input.txt").

6-It performs reconstruction starting from short reads obtained from RNA-seq. It is best suited to manage the transcriptomes of lower eukaryotes with a low number of introns per gene (usually zero or one per gene) and it can be used also for prokaryotes.
Policistronic transcripts are reported in the output file both for eukaryotes (where they are very rare) and for prokaryotes and can be easily found in the output file "final_prediction.gff" because the transcript ID has two different protein-coding genes IDs. Policistronic transcripts were found only when two adjacent genes on the same strand are "included" into the same transcript and when they have similar coverage on the two protein-coding regions (less than 2 fold coverage difference).

ORA needs a sam file with the reads aligned on the reference genome and (optional) a gff file reporting the annotation and the position of genes on the genome (a test file is available "saccharomyces_cerevisiae_annotation_test_file.gff").

**The main output files** report the positions of the transcripts identified in the genome with their length and coverage, the transcripts IDs based on the genes present in the annotation ".gff" file (final_prediction.gff) and the UTR size (all_UTR).

## DOWNLOADING AND INSTALLATION

The refence-based transcriptome reconstruction software ORA is composed by a bash script and four perl scripts:
**"launcher.pl"**
The launcher is a bash script that runs the single parts of the software, it simplify the procedure.
The other four scripts: **"overlapping_reads_assembler_v2.pl"** **"seconda_parte.pl"** **"terza_parte.pl"** **"quarta_parte.pl"**
MUST BE in the same folder where the "launcher.pl" was saved.
The script **"quarta_parte.pl"** determines the slope of the 5'-UTR terminal regions of the transcripts and it helps the identification of alternative transcription start and termination sites (TSS, TTS).

After the download of the various scripts IN THE SAME FOLDER, make them executable using chmod command.
Assume the scripts were downloaded in folder named "folder"
user@machine:~/chmod 755 path/to/folder/launcher.pl
user@machine:~/chmod 755 path/to/folder/overlapping_reads_assembler_v2.pl
user@machine:~/chmod 755 path/to/folder/seconda_parte.pl
user@machine:~/chmod 755 path/to/folder/terza_parte.pl
user@machine:~/chmod 755 path/to/folder/quarta_parte.pl
user@machine:~/chmod 755 path/to/folder/estrapola_chr.pl

## MANDATORY AND OPTIONAL FILES

In order to work properly, ORA needs three files:

**(1)-** a file with the reads mapped on the reference genome in sam format, the file has to be ordered using the samtools (Li et al., 2009) (http://samtools.sourceforge.net/) (an example file is provided "saccharomyces_cerevisiae_3Mreads_ordered.sam.gz");
A few lines of the SAM file are reported below:

```
cerevisiae_rep_1.7447889 solid0034_20090925_FC1_JGYeast_507_2002_1682 length=35  0       chr01   215     35      35M
        *       0       0ACCGTTACCCTCCAATTACCCATATCCAACCCACT     =<;==9::9;<<<<<:9:<<<=<::9:86:76:;<
        MD:Z:35 NM:i:0  X1:Z:1_35_S1E34A25L34_T3Q25_BS1  HI:i:1IH:i:1
```

cerevisiae_rep_1.5898722 0 chr01 1117 23 13M402N10M * 0 0
AGACGATACTGTGATTTCCAATT )0.'&+190$,440'(490&&19 MD:Z:23 NM:i:0 X2:Z:[AT..AC] X3:f:2
X4:f:0.0004 IH:i:1
cerevisiae_rep_1.33834674 16 chr01 1140 21 13M1623N8M * 0 0
TATTTAATAGGATGGATGGTA 3-('$#(++**/+9<;:7762 MD:Z:21 NM:i:0 X2:Z:[CT..GC] X3:f:5
X4:f:0.0247 IH:i:1
cerevisiae_rep_1.14271585 0 chr01 1196 17 11M1262N6M * 0 0
AGCGGCGTATAAGAATG 687362045645944/& MD:Z:17 NM:i:0 X2:Z:[CT..GC] X3:f:5 X4:f:0.2671
IH:i:3
cerevisiae_rep_1.5849818 solid0034_20090925_FC1_JGYeast_400_1721_1293 length=35 16 chr01 1807 34 35M
* 0 0CTAGTNTGCGATAGTGTAGATACCGTCCTTGGATA >>>;<<;===?<8;<;:8:::<>=55<;3'*1685
MD:Z:35 NM:i:0 X1:Z:35_1_S1E34A24L34_T0Q30_BS1 HI:i:5IH:i:10
cerevisiae_rep_1.12676941 solid0034_20090925_FC1_JGYeast_858_1992_548 length=35 16 chr01 1807 35 35M
* 0 0CTAGTTTGCGATAGTGTAGATACCGTCCTTGGATA 9:<:::99;;;;<??>=<=:9>>=97==8.2::=<
MD:Z:35 NM:i:0 X1:Z:35_1_S1E34A25L34_T0Q23_BS1 HI:i:7IH:i:10
cerevisiae_rep_1.18230516 solid0034_20090925_FC1_JGYeast_1232_1125_2015 length=35 16 chr01 1807 33
35M * 00 CTAGTNTGCGNTAGTGTAGATACCGTCCTTGGATA
844:;71144.1514562401=:812:6.(.24:; MD:Z:35 NM:i:0 X1:Z:35_1_S1E34A19L34_T0Q21_BS1HI:i:6 IH:i:10
cerevisiae_rep_1.2328659 solid0034_20090925_FC1_JGYeast_165_258_875 length=35 16 chr01 1814 35 35M
* 0 0GCGATAGTGTAGATACCGTCCTTGGATAGAGCACT
<;<=<?@>>?====>@><>>?><;=><;;=>44== MD:Z:35 NM:i:0 X1:Z:35_1_S1E34A27L34_T3Q25_BS1 HI:i:5IH:i:9
cerevisiae_rep_1.6586352 solid0034_20090925_FC1_JGYeast_450_536_92 length=35 16 chr01 1814 35 35M
* 0 0GCGATAGTGTAGATACCGTCCTTGGATAGAGCACT :8:===<<<;;;;;;;;;:5523::703;;;00;<
MD:Z:35 NM:i:0 X1:Z:35_1_S1E34A24L34_T3Q27_BS1 HI:i:5IH:i:9
cerevisiae_rep_1.6851993 solid0034_20090925_FC1_JGYeast_467_1136_1246 length=35 16 chr01 1814 35 35M
* 0 0GCGATAGTGTAGATACCGTCCTTGGATAGAGCACT 9:97:<?A>9:<=:9==<:9<>;;=?:89:=87;;
MD:Z:35 NM:i:0 X1:Z:35_1_S1E34A26L34_T3Q23_BS1 HI:i:5IH:i:9

**(2)-** a file with chromosomes names and their lengths (an example file is provided "saccharomyces_cerevisiae_S288c_input.txt");
An example of the file structure is reported below

| | |
|---|---|
| chr01 | 230208 |
| chr02 | 813178 |
| chr03 | 316617 |
| chr04 | 1531918 |
| chr05 | 576869 |
| chr06 | 270148 |
| chr07 | 1090947 |
| chr08 | 562643 |
| chr09 | 439885 |
| chr10 | 745745 |
| chr11 | 666454 |
| chr12 | 1078175 |
| chr13 | 924429 |
| chr14 | 784333 |
| chr15 | 1091289 |
| chr16 | 948062 |
| Mit | 85779 |
| 2micron | 6318 |

**(3)-** a gff file with the genome annotation; the structure of the gene features reported in this file is identical to that reported in the "gff" file present in the SGD database (an example file is provided "saccharomyces_cerevisiae_annotation_test_file.gff");
a few lines of the file are reported below:

chr01 SGD chromosome 1 230218 . . .
ID=chr01;dbxref=NCBI:NC_001133;Name=chr01
chr01 SGD telomeric_repeat 1 62 . - . ID=TEL01L-TR;Name=TEL01L-
TR;Note=Terminal%20stretch%20of%20telomeric%20repeats%20on%20the%20left%20arm%20of%20Chromosome%20I;display=Terminal%20telomeric%20repeats%20on%20the%20left%20arm%20of%20Chromosome%20I;dbxref=SGD:S000028864
chr01 SGD telomere 1 801 . - .
ID=TEL01L;Name=TEL01L;Note=Telomeric%20region%20on%20the%20left%20arm%20of%20Chromosome%20I%3B%20composed%20of%20an%20X%20element%20core%20sequence%2C%20X%20element%20combinatorial%20repeats%2C%20and%20a%20short%20terminal%20stretch%20of%20telomeric%20repeats;display=Telomeric%20region%20on%20the%20left%20arm%20of%20Chromosome%20I;dbxref=SGD:S000028862
chr01 SGD X_element_combinatorial_repeat 63 336 . - . ID=TEL01L-
XR;Name=TEL01L-
XR;Note=Telomeric%20X%20element%20combinatorial%20Repeat%20region%20on%20the%20left%20arm%20of%20Chromosome%20I%3B%20contains%20repeats%20of%20the%20D%2C%20C%2C%20B%20and%20A%20types%2C%20as%20well%20as%20Tbf1p%20binding%20sites%3B%20formerly%20called%20SubTelomeric%20Repeats;display=Telomeric%20X%20element%20Repeat%20region%20on%20the%20left%20arm%20of%20Chromosome%20I;dbxref=SGD:S000028866
chr01 SGD gene 335 649 . + .
ID=YAL069W;Name=YAL069W;Ontology_term=GO:0003674,GO:0005575,GO:0008150;Note=Dubious%20open%20reading%20frame%20unlikely%20to%20encode%20a%20protein%2C%20based%20on%20available%20experimental%20and%20comparative%20sequence%20d

ata;display=Dubious%20open%20reading%20frame%20unlikely%20to%20encode%20a%20functional%20protein;dbxref=SGD:S000002143;orf_classification=Dubious
chr01 SGD CDS 335 649 . + 0 Parent=YAL069W;Name=YAL069W_CDS;orf_classification=Dubious
chr01 SGD X_element 337 801 . - . ID=TEL01L-XC;Name=TEL01L-XC;Note=Telomeric%20X%20element%20Core%20sequence%20on%20the%20left%20arm%20of%20Chromosome%20I%3B%20contains%20an%20ARS%20consensus%20sequence%2C%20an%20Abf1p%20binding%20site%20consensus%20sequence%20and%20two%20small%20overlapping%20ORFs%20(YAL068W-A%20and%20YAL069W);display=Telomeric%20X%20element%20Core%20sequence%20on%20the%20left%20arm%20of%20Chromosome%20I;dbxref=SGD:S000028865
chr01 SGD nucleotide_match 753 763 . - . ID=TEL01L-XC_nucleotide_match;Name=TEL01L-XC_nucleotide_match;dbxref=SGD:S000028865
chr01 SGD binding_site 532 544 . - . ID=TEL01L-XC_binding_site;Name=TEL01L-XC_binding_site;dbxref=SGD:S000028865
chr01 SGD gene 538 792 . + . ID=YAL068W-A;Name=YAL068W-A;Ontology_term=GO:0003674,GO:0005575,GO:0008150;Note=Dubious%20open%20reading%20frame%20unlikely%20to%20encode%20a%20protein%3B%20identified%20by%20gene-trapping%2C%20microarray-based%20expression%20analysis%2C%20and%20genome-wide%20homology%20searching;display=Dubious%20open%20reading%20frame%20unlikely%20to%20encode%20a%20functional%20protein;dbxref=SGD:S000028594;orf_classification=Dubious
chr01 SGD CDS 538 792 . + 0 Parent=YAL068W-A;Name=YAL068W-A_CDS;orf_classification=Dubious
chr01 SGD ARS 650 1791 . . . ID=ARS102;Name=ARS102;Alias=ARSI-1;Note=Autonomously%20Replicating%20Sequence;display=Autonomously%20Replicating%20Sequence;dbxref=SGD:S000121252
chr01 landmark region 24000 27968 . - . ID=FLO9
chr01 SGD gene 24000 27968 . - . ID=YAL063C;Name=YAL063C;gene=FLO9;Alias=FLO9;Ontology_term=GO:0000128,GO:0000501,GO:0005537,GO:0009277;Note=Lectin-like%20protein%20with%20similarity%20to%20Flo1p%2C%20thought%20to%20be%20expressed%20and%20involved%20in%20flocculation;display=Lectin-like%20protein%20with%20similarity%20to%20Flo1p;dbxref=SGD:S000000059;orf_classification=Verified
chr01 SGD tRNA 139152 139254 . + . ID=tP(UGG)A;Name=tP(UGG)A;gene=TRN1;Alias=TRN1;Ontology_term=GO:0005829,GO:0006414,GO:0030533;Note=tRNA-Pro%3B%20target%20of%20K.%20lactis%20zymocin;display=tRNA-Pro;dbxref=SGD:S000006680
chr01 SGD intron 139188 139218 . + . Parent=tP(UGG)A;Name=tP(UGG)A_intron;dbxref=SGD:S000006680
chr01 SGD noncoding_exon 139152 139187 . + . Parent=tP(UGG)A;Name=tP(UGG)A_noncoding_exon;dbxref=SGD:S000006680
chr01 SGD noncoding_exon 139219 139254 . + . Parent=tP(UGG)A;Name=tP(UGG)A_noncoding_exon;dbxref=SGD:S000006680
chr12 SGD rRNA 451575 458432 . - . ID=RDN37-1;Name=RDN37-1;gene=RDN37-1;Alias=RDN37-1,RDN37;Note=35S%20ribosomal%20RNA%20transcript%2C%20encoded%20by%20the%20RDN1%20locus%2C%20that%20is%20processed%20into%20the%2025S%2C%2018S%20and%205.8S%20rRNAs%20(represented%20by%20the%20RDN25%2C%20RDN18%2C%20and%20RDN58%20loci);display=35S%20ribosomal%20RNA%20transcript;dbxref=SGD:S000006486
chr12 SGD external_transcribed_spacer_region 451575 451785 . - . Parent=RDN37-1;Name=RDN37-1_external_transcribed_spacer_region;dbxref=SGD:S000006486

**The relevant fields (separated by "tab" spacers) are:**

1-the chromosome name - it is mandatory that names of the chrs/scaffolds are IDENTICAL than those reported in the a file with chromosomes names and their lengths (see above);
2-the method followed for gene annotation (SGD, manual,…);
3-features, VERY IMPORTANT, only features named "gene", "tRNA", "snoRNA" and "snRNA" are considered by the software, those having different names are discarded;
4-transcript start;
5-transcript end;
6-a "." character;
7-the strand ("+" or "-"), it is VERY IMPORTANT;
8-a "." character;
9-some characteristics separated by ";" characters - the "ID=name" field is MANDATORY because it defines the name of the gene(s) that will be assigned to the transcripts in the output files. In the gene names please do not use characters like ";" or other characters. tRNAs must follow the standard nomenclature ID=tF(GAA)B the "t" at the beginning in lowercase letter.

If the "gff" file with the genome annotation is not available, ORA stops after the first transcriptome reconstruction step and it provides as output only the positions of the transcripts (blocks) that it was able to reconstruct, but usually the transcripts are highly fragmented and (obviously) they cannot be referred to the annotated genes.

**PRELIMINARY STEPS, HOW TO ALIGN SEQUENCES ON THE REFERENCE GENOME**

Before transcriptome reconstruction process, reads have to be aligned on the reference genome using short-reads alignment software and the output file in SAM format has to be ordered. This procedure can be afforded using a

fast splice junction mapper for RNA-Seq reads like TopHat (Trapnell et al., 2009) or PASS (Campagna et al., 2009).
The SAM output obtained from this step has to be ordered using SAM tools, we provide a small bash script to manage this procedure (sam_sorter.pl).
sam_sorter.pl needs as input only a compressed sam file (sam_file.gz) and, prior to use the script, it is necessary to set the correct path to the folder were sam tools were saved (lines 21-23-25-27).
user@machine:~/$ perl ./sam_sorter.pl path/to/the/sam_file.gz
Alternatively the user can directly run the sam tools "view" and "sort".

## EXAMPLE - HOW TO PERFORM THE TRANSCRIPTOME PREDICTION

After the short-reads alignment process and the "order sam" step, you are ready for the last step.
We assume you have changed your working directory moving into the folder containing the software (using the "cd" command);

**To run the software use the following command line**

**launcher.pl followed by three arguments: the sam file, the file with chromosome names and size and the gff file with the annotation (optional) (see below)**

user@machine:~/folder$ ./launcher.pl file_sam_odered.sam chroms_name_and_size.txt annotation_file.gff

After launching the software, user has to set some parameters, if you do not have clear in mind how to set the parameters values, please use the default. In grey are highlighted the parameters required by the software:

1-the user can select only unique reads of the sam file for transcriptome reconstruction (those univocally aligned) or both the unique and the multiple aligned reads:
-- Do you want to use unique reads only? (y/n) --

2-The user can select the minimum number of reads assembled into a single "block", this is very important to reduce the background, since a lot of reads are "singletons" aligned into the genome in random positions;
-- Insert minimum number of reads to define a block (default 5) --

3- The user can select the maximum distance between transcripts to consider them as independent transcripts, this is very important because some internal regions of the transcripts can have "zero coverage" and frequently this effect is due to biases on the RNA-seq process; transcripts closer than selected threshold will be joined in a single transcript;
-- Insert maximum distance between reads allowed to merge them in a single block. (default 20) --

4-The software calculates the number of reads in the input file, the number of unique reads and the number of reads per chromosome.
The software assembles the transcript "blocks"; for genes having low coverage and for genes having regions that are difficult to sequence, these blocks do not cover the entire transcript and they can be joined in the subsequent step to obtain a better prediction of the transcript structure.
If the annotation file was not provided to the software, it will complete the process at this step.

5-If the annotation file was provided, the software can assign the vast majority of the transcripts to the genes reported in the annotation file and can also identify the transcripts that cannot be assigned to any annotated gene.

6-The software reports the total number of transcribed regions (blocks) located into the same transcript that were joined during assembly and also the mean and standard deviation of the size of all the blocks localized into the same gene:

Number of intergenic blocks joined together: 1174
--- Mean size: 170.237
--- SD size: 199.017

The software reports the size distribution of the transcripts generated at this step, this will be useful to select in the next step the minimum size of the transcripts that will be reported in the output. An example is given below

and shows that ORA identify 321 transcript blocks with size between 0 and <=50 bases, 969 with size between 50 and <=150 bases, etc..

```
Frequence distribution for block's size
|-------------------------------------------------------------------------------|         |0 -- 50 -> 739 features
|-----------------------------------------------------------------------------------------------------------|         |50 -- 100 -> 969
features
|-------------------------------------------------------------------------------|         |100 -- 150 -> 698 features
|---------------------------------|         |150 -- 200 -> 321 features
|-----------------|         |200 -- 250 -> 164 features
|-------------|         |250 -- 300 -> 125 features
|-----------|         |300 -- 350 -> 105 features
|----------|         |350 -- 400 -> 98 features
|--------|         |400 -- 450 -> 72 features
|----|         |450 -- 500 -> 41 features
|----|         |500 -- 550 -> 40 features
|----|         |550 -- 600 -> 41 features
|--|         |600 -- 650 -> 24 features
|--|         |650 -- 700 -> 25 features
|-|         |700 -- 750 -> 15 features
|-|         |750 -- 800 -> 10 features
|-|         |800 -- 850 -> 17 features
||         |850 -- 900 -> 6 features
||         |900 -- 950 -> 8 features
||         |950 -- 1000 -> 7 features
|-|         |1000 -- 1050 -> 9 features
||         |1050 -- 1100 -> 8 features
||         |1100 -- 1150 -> 7 features
||         |1150 -- 1200 -> 4 features
||         |1200 -- 1250 -> 2 features
||         |1250 -- 1300 -> 3 features
||         |1300 -- 1350 -> 3 features
||         |1350 -- 1400 -> 1 features
||         |1400 -- 1450 -> 4 features
||         |1450 -- 1500 -> 0 features
||         |1500 -- 1550 -> 1 features
||         |1550 -- 1600 -> 1 features
||         |1600 -- 1650 -> 0 features
||         |1650 -- 1700 -> 1 features
```

The software links all the “blocks” that are not separated by introns predicted from the spliced reads and that are localized into the same gene; this reduces the “fragmentation” of the transcripts.

7- The user can select the minimum size of the blocks that will be reported in the output file:

```
-- Insert minimum intergenic blocks' size. (default 170.237) --

Genes without reference deleted = 1245 (on the base of the coverage 203) of 1507
Chrs checked |==================| Done
Genes moved in array "other" : 25
Chrs checked |==================| Done
Final refinement of the terminal regions removes very low coverage regions:
3 - 1 - Refinement of predicted genes with mean coverage in their terminal parts higher than 20.
Chrs checked |==================| Done
Refined genes: 481
3 - 2 - Calculation of the new mean coverage for each gene
Chrs checked |==================| Done
Do you want to perform analysis of putative alternative UTRs and slope? (y/n)
y
4 - 1 - 5' and 3' UTRs analysis
```

Chrs checked |==================| Done

8-The software performs the final calculations, the user can chose between the generation of an output file for each chromosome or of a single global file; the script "estrapola_chr.pl" separate the results into ".gff" files, one for each chromosome that can be uploaded in Artemis software (Rutherford et al., 2000) for a manual check.

Do you want to produce gff output for each chromosome? (y/n)

9-Finally, the software can perform a final analysis to check for the presence of putative alternative UTRs and to check the slope of the 5'-UTR coverage (this part was not validated in the present version of the software).
Do you want to perform analysis of putative alternative UTRs and slope? (y/n)

## OUTPUT FILES

Output files are saved in a folder having the name assigned to the assembly project by the user at the beginning of the process and they are localized in the same folder of the software.

Into the folder named **"coverage"**, are reported the coverage files separated for each chromosome; these files can be visualized with Artemis (Rutherford et al., 2000) (http://www.sanger.ac.uk/resources/software/artemis/) using the command "Graph" "Add user plot";

"unrefinedblocks.gff" is an intermediate file in the transcriptome reconstruction process containing the transcripts assembled without the support of the reference annotation file;
"histogram_blocks_noref.gff" contains the size distribution of the transcript blocks assembled at the first step (without the support of the reference file);
"final_prediction.gff" **is THE MAIN OUTPUT FILE** with the transcripts in "gff" format assembled considering the information in the ".gff" annotation file;
"all_UTR" contains the size of the 5'- and 3'-UTR regions calculated considering the transcripts identified and the protein-coding regions reported in the "gff" file;
"unrefinedblocks.stat" reports the statistics of the shotgun reads used for assembly, the number of "unique" reads (aligned only in one genomic position), the number of "not unique" reads, etc..

The files "slope_5UTR", "cluster_3UTR", "cluster_5UTR" report some information on the 5'- and 3'-UTR coverage for the transcripts having a coverage higher than 20 and an UTR size equal or higher than 100 bp. These files are a representation of the coverage distribution around the transcription start site (TSS).
It is known that several genes have alternative UTRs. Analysis of the coverage profile can help the identification of the genes having a sudden and stepwise increase of the coverage in the UTR region, these can represent alternative UTRs.
During this task, ORA splits the UTR regions in four equal parts, it calculates the mean coverage for each one and calculates the ratio between the coverage of each part and the average coverage of the ORF producing four "ratio values". As a positive control, it calculates a fifth ratio from a window inside the ORF with a size equal to the other four. This fifth ratio is a "control" that helps to determine if the coverage modification could be due to an uneven profile or to a real alternative TSS or TTS.
For each gene and each step the output is reported into the files "cluster_3UTR" and "cluster_5UTR", where are reported the gene name, mean coverage, size of UTR and values of the five windows (from the outermost to the innermost of the transcript). Ratios close to 1 indicate similar mean coverage on adjacent windows. Output can be analyzed using software like MeV (Saeed et al., 2003) (http://www.tm4.org/mev.html) to generate clusters of transcripts having similar UTR coverage "behaviors".
ORA also analyzes the coverage profile around the TSS to determine if the profile reaches a value similar to the mean coverage level of the gene within the first 30 bases. ORA calculates for each base starting from position "1" to "30" the coverage increment normalized with respect to the mean coverage of the gene using the formula:
$Slope = (x_{analyzed} - x_{before}) / (ORF\ mean\ coverage\ value)$
where "$x_{analyzed}$" is the coverage at the "current position" and "$x_{before}$" is the coverage one base upstream. The output file named "slope_5UTR" shows the gene name, mean coverage, the first and last base position analyzed and finally the thirty ratios calculated using the formula. Output can be analyzed using MeV software (Saeed et al., 2003).

**REFERENCES**


Campagna D, Albiero A, Bilardi A, Caniato E, Forcato C, Manavski S, Vitulo N, Valle G. **PASS: a program to align short sequences.** Bioinformatics. 2009 1;25(7):967-8.

Langmead B, Trapnell C, Pop M, Salzberg SL. **Ultrafast and memory-efficient alignment of short DNA sequences to the human genome.** Genome Biol. 2009;10(3):R25.

Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R; 1000 **Genome Project Data Processing Subgroup. The Sequence Alignment/Map format and SAMtools.** Bioinformatics. 2009 Aug 15;25(16):2078-9.

Rutherford K, Parkhill J, Crook J, Horsnell T, Rice P, Rajandream MA, Barrell B. **Artemis: sequence visualization and annotation.** Bioinformatics. 2000 16(10):944-5.

Trapnell C, Pachter L, Salzberg SL. **TopHat: discovering splice junctions with RNA-Seq.** Bioinformatics. 2009 May 1;25(9):1105-11.

Saeed AI, Sharov V, White J, Li J, Liang W, Bhagabati N, Braisted J, Klapa M, Currier T, Thiagarajan M, Sturn A, Snuffin M, Rezantsev A, Popov D, Ryltsov A, Kostukovich E, Borisovsky I, Liu Z, Vinsavich A, Trush V, Quackenbush J. **TM4: a free, open-source system for microarray data management and analysis.** Biotechniques. 2003 Feb;34(2):374-8.